Ilona Škarydová; Milan Hokr

Solution of mechanical problems in fractured rock with the user-defined interface of COMSOL multiphysics

In: Jan Chleboun and Petr Přikryl and Karel Segeth and Jakub Šístek and Tomáš Vejchodský (eds.): Programs and Algorithms of Numerical Mathematics, Proceedings of Seminar. Dolní Maxov, June 8-13, 2014. Institute of Mathematics AS CR, Prague, 2015. pp. 200--206.

Persistent URL: http://dml.cz/dmlcz/702685

Programs and Algorithms of Numerical Matematics 17
J. Chleboun, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2015

SOLUTION OF MECHANICAL PROBLEMS IN FRACTURED ROCK WITH THE USER-DEFINED INTERFACE OF COMSOL MULTIPHYSICS

Ilona Škarydová[1], Milan Hokr[2]

[1] Faculty of Mechatronics, Informatics and Interdisciplinary Studies,
Technical University of Liberec
Studentská 1402/2, 461 17 Liberec 1, Czech Republic
ilona.skarydova@tul.cz
[2] Institute for Nanomaterials, Advanced Technologies and Innovation,
Technical University of Liberec
Studentská 1402/2, 461 17 Liberec 1, Czech Republic
milan.hokr@tul.cz

Abstract

This paper presents the main concept and several key features of the user-defined interface of COMSOL Java API for the solution of mechanical problems in fractured rock. This commercial computational system based on FEM has yet to incorporate fractures in mechanical problems.

Our aim is to solve a 2D mechanical problem with a fracture which is defined separately from finite-element discretization and the fracture properties are included through the constitutive laws. This will be performed based on the principles of the extended finite element method as a way of fracture description, enrichment functions for rock elements containing fractures, etc.

We present an approach to describing a simple mechanical problem in COMSOL Java API together with a proposal of a solution method, and we also demonstrate the potential of COMSOL Java API for solving more complicated problems with fractures.

1. Introduction

For many applications it is important to evaluate the effects of fractures in rock on its mechanical properties and the effects of stress on fracture opening/closure. An example which is also the context of our work is the concept of the geological disposal of spent nuclear fuel, with three protective barriers (copper/steel containers, bentonite, and a stable rock massif). Safety analysis requires an understanding and prediction of the complex thermo-hydro-mechanical-chemical processes and the current engineering software can sometimes lack the required detail.

Despite the fact that a number of methods for solving problems of fractured rock mechanics (DEM [2], XFEM [7], FEM with special fracture elements [5]) are implemented in various software packages, most are too specialized and have complicated coupling with other important processes.

In this paper we present a methodology for solving specialized problems using a commercially available code programming interface, combining the advantages of established supported code with the freedom of an open-source project. The use of Java API in COMSOL Multiphysics software is presented here. Firstly, we describe the implementation process of a simple plane strain problem, then we highlight several advanced functions and other required features (constitutive laws, XFEM principles) of COMSOL Java API.

2. Description of a plane strain problem

The approach will be described using a basic 2D plane strain problem which is represented by Hooke's law

$$\begin{pmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{pmatrix} = \frac{E}{(1+\nu)(1-2\nu)} \begin{pmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & 1-2\nu \end{pmatrix} \begin{pmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{pmatrix}, \tag{1}$$

where σ_x, σ_y and τ_{xy} are normal and shear components of the stress tensor, E is Young's modulus, ν is Poisson's ratio, ε_x, ε_y and γ_{xy} are normal and shear components of the strain tensor and the equilibrium equations are written as

$$\begin{aligned} \frac{\partial \sigma_x}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} + k_x &= 0 \\ \frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \sigma_y}{\partial y} + k_y &= 0, \end{aligned} \tag{2}$$

where k_x and k_y are components of a force vector.

The dependence of the strain tensor components on components of the displacement vector $\mathbf{u} = [u, v]^T$ for a small deformation is expressed by

$$\varepsilon_x = \frac{\partial u}{\partial x} \qquad \varepsilon_y = \frac{\partial v}{\partial y} \qquad \varepsilon_{xy} = \frac{1}{2}\left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right) = \frac{1}{2}\gamma_{xy}. \tag{3}$$

If we substitute Hooke's law (1) together with the strain-displacement relation (3) into the equilibrium equations (2), we get the form

$$\begin{aligned} \frac{\partial}{\partial x}\left[\left(\lambda + 2\mu\right)\frac{\partial u}{\partial x} + \lambda \frac{\partial v}{\partial y}\right] + \frac{\partial}{\partial y}\left[\mu\left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right)\right] + k_x &= 0 \\ \frac{\partial}{\partial x}\left[\mu\left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right)\right] + \frac{\partial}{\partial y}\left[\lambda \frac{\partial u}{\partial x} + \left(\lambda + 2\mu\right) + \frac{\partial v}{\partial y}\right] + k_y &= 0, \end{aligned} \tag{4}$$

where λ and μ are so-called Lamé coefficients, which can be derived from Young's modulus and Poisson's ratio. This form can be used for specific expression of solved problem in a "General form PDE" in COMSOL Multiphysics.

3. Computational tool

The computational system COMSOL Multiphysics based on FEM was chosen as an appropriate computational tool. Despite the fact that features for calculating with fractures in mechanical problems are not included in it, COMSOL provides sufficient variability for their implementation. In addition it has proven to be a suitable tool for solving coupled processes.

We use a special interface of COMSOL Multiphysics called the Java application programming interface (COMSOL Java API, [1]), which provides access to special extended features and functions that are not available from commonly used graphical user interface (GUI).

The basic model is possible to define and export from GUI (model.java) or directly define in the integrated development environment Eclipse, [3] using the appropriate commands. Then the model in Java code is processed in Eclipse. This environment also allows new results to be directly exported in different formats (.jpg, .png, .txt, .cls) or the access and connection with the GUI of COMSOL Multiphysics. A diagram of the approach to the solution using COMSOL is summarized in Figure 1.

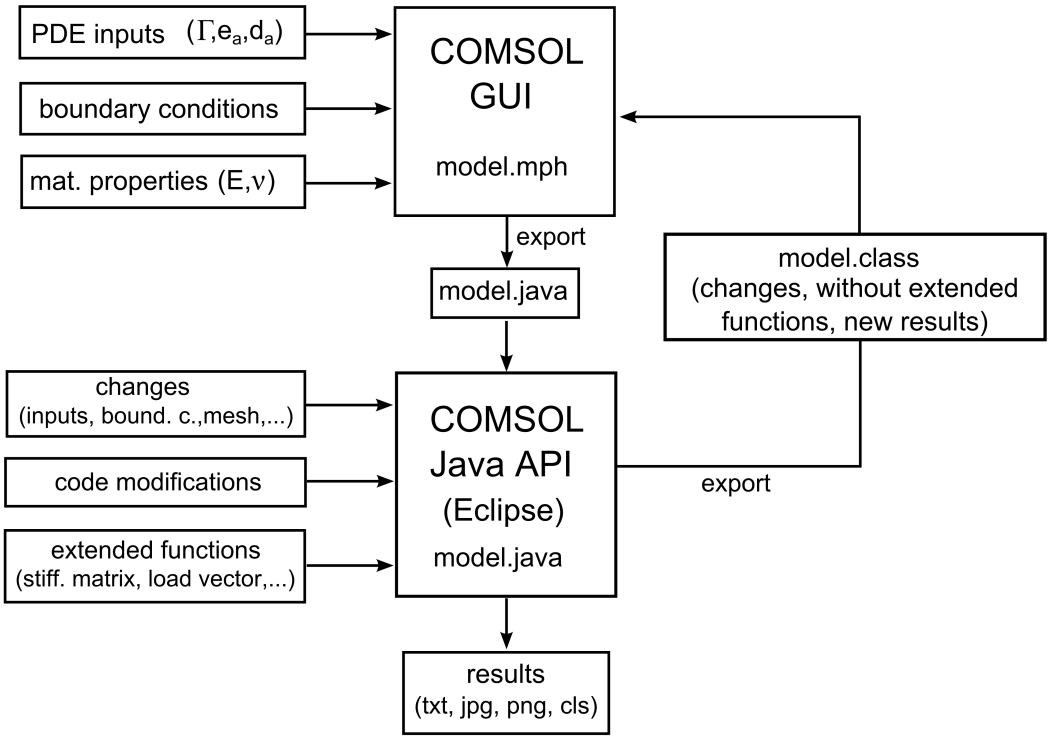

Figure 1: Approach of the solution using COMSOL Java API, solved using a user-defined PDE with Hooke's law (1) as a constitutive relation

The built-in physical interface which describes the physical process using a partial differential equation is replaced by a "Mathematical module" with a "General form PDE" interface. Within this module it is possible to apply a user-defined partial differential equation through the individual coefficients. The form of the partial differential equation is specified as

$$e_a \frac{\partial^2 \mathbf{u}}{\partial t^2} + d_a \frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \Gamma = f \qquad \mathbf{u} = [u, v]^T, \tag{5}$$

where $\mathbf{u}$ is a displacement vector, e_a is a mass coefficient, d_a is a damping coefficient and matrix Γ can be expressed by

$$\begin{aligned} \Gamma_{1x} = (\lambda + 2\mu)\frac{\partial u}{\partial x} + \lambda \frac{\partial v}{\partial y}, \qquad \Gamma_{1y} = \mu\left(\frac{\partial u}{\partial y} + \lambda \frac{\partial v}{\partial x}\right), \\ \Gamma_{2x} = \Gamma_{1y} \qquad \Gamma_{2y} = \lambda \frac{\partial u}{\partial x} + (\lambda + 2\mu)\frac{\partial v}{\partial y}\Big), \end{aligned} \tag{6}$$

which is taken from (4). For a steady-state problem the last divergence term on the left side of the equation (5) and $e_a = 0, d_a = 0$ are considered.

The results of the model can be displayed using exported model.class in the GUI of COMSOL Multiphysics or in a simply created graphical interface in COMSOL Java API. Unfortunately, exported model with Java modifications is unable to store results for recomputed solution in the GUI (extended functions are not included in GUI).

4. Perspectives for fracture mechanics implementation

The previous section described the elementary plane strain problem and how to define and solve it in COMSOL Java API with a user-defined PDE. The following section specifies other important aspects and features (i.e. fractures with a predetermined fixed position and their influence on the elastic properties of rock, constitutive laws, special functions of COMSOL Java API, use of certain XFEM principles) which are necessary for defining the problem with the fractures.

4.1. Constitutive laws

Constitutive laws are special empirical or theoretical formulas which express the behaviour of a material under a general load. They are important for describing the rock-matrix behaviour and also for expressing the influence of a fracture on rock mass properties. A large number of constitutive laws are referred to in [6]: different relations are proposed for rock and fractures. Constitutive laws are often described in terms of "strength-form" (they describe the limit stress when the failure occurs) but for our purpose the stress-strain relation is more suitable.

The simplest and the most popular formula for expressing elastic rock behaviour is the above-mentioned Hooke's law (1) with two independent material parameters

for an isotropic material. The elastic/plastic behaviour of the rock can be described using the Mohr-Coulomb or Hoek-Brown constitutive laws, which assume that the failure occurs under the highest shear stress.

The behaviour of the fractures can be described for example by empirical criteria (e.g. the Mohr-Coulomb criterion, Goodman's law, the Barton-Bandis criterion) or theoretical models (the Amadei-Saeb or Plesha model). For example, the Barton-Bandis model is an empirical constitutive model requiring JRC (joint roughness coefficient) and JCS (joint wall compressive strength) parameters.

4.2. Fractures and their representation

Fractures and their representation in the model are the next important part of the implementation. COMSOL Multiphysics does not have any built-in approach to solving mechanical problems with fractures. Thus, fractures have to be controlled externally in Java code.

Fractures are represented by lines in the 2D case and are defined separately (they are not dependent on the computational mesh). A similar approach is used in the XFEM family of methods [4]. Many of them use the so-called level-set method [8], which does not require curve parameterization and is also more suitable for problems with moving interfaces or growing fractures.

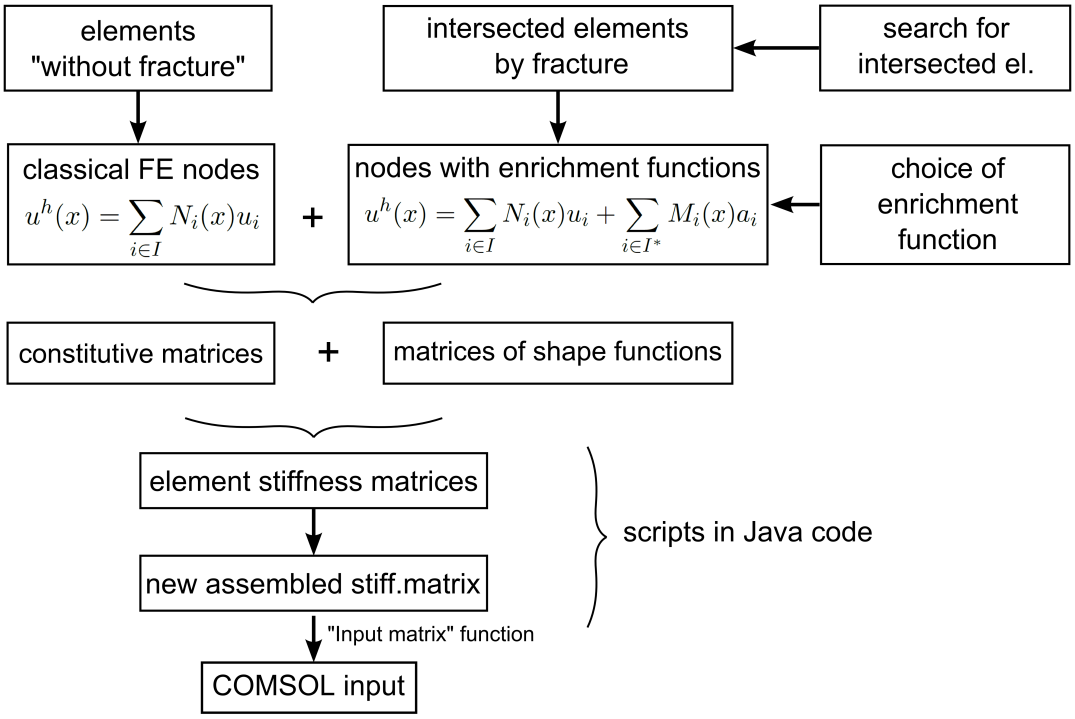

Figure 2: XFEM solution for fractures in COMSOL Java API

4.3. Implementation and related problems

We can then apply several principles of the Extended Finite Element method ([7, 4]) to the problem with fractures. XFEM is a numerical method based on a generalization of FEM and it enables a local enrichment of approximation spaces (hence discontinuous behaviour in a small part of the domain). In this way it is possible to define both weak and strong discontinuities (discontinuity in the stress and strain field or discontinuity in the displacement field, respectively).

For the implementation of XFEM it is necessary to use special functions available in COMSOL Java API (see Figure 2). One of these functions is the "Input matrix", which enables the input of an externally assembled matrix or vector (stiffness, mass or damping matrix and load vector) in a sparse form. One disadvantage is that the matrices are not stored in the model, which has to be processed externally.

The next problem, which is necessary to deal with it, is the detection of intersection elements (elements of the rock matrix which are intersected by the fracture). This feature can be solved by code in COMSOL Java API using the known position of the fracture and the positions of the individual elements from the exported mesh file.

5. Conclusions

We have shown that the COMSOL Java API can be conveniently used for testing or applying new modelling concepts and numerical schemes, which can be of interest to the wider community.

The example of including fractures to the elasticity problem has been presented on a conceptual level, with its implementation planned for future work.

Acknowledgements

This work was supported by the project LO1201 through the financial support of the Ministry of Education, Youth and Sports in the framework of the targeted support of the National Programme for Sustainability I, by the Ministry of Education of the Czech Republic within the SGS project no. 21066/115 on the Technical University of Liberec and by the Ministry of Industry and Trade of the Czech Republic within the project FR TI3/579.

References

[1] COMSOL AB, Stockholm, Sweden: *Comsol Java API reference guide*, 1^{st} ed.,2012. Version 4.3a.

[2] Cundall, P. A. and Strack, O. D. L.: A discrete numerical model for granular assemblies. Géotechnique **29** (1979), 47–65.

[3] Eclipse: integrated development environment. `<https://www.eclipse.org/>`. Accessed: 2014-09-10.

[4] Fries, T. P. and Belytschko, T.: The extended/generalized finite element method: An overview of the method and its applications. Internat. J. Numer. Methods Engrg. **84.3** (2010), 253–304.

[5] Goodman, R. E.: *Methods of geological engineering in discontinuous rocks*. West Publishing Company, San Francisco, CA, 1976.

[6] Jing, L. and Stephansson, O.: *Fundamentals of discrete element methods for rock engineering: theory and applications*. Elsevier, Amsterdam, 2007.

[7] Moës, N., Dolbow, J., and Belytschko, T.: A finite element method for crack growth without remeshing. Internat. J. Numer. Methods Engrg. **46** (1999), 131–150.

[8] Osher, S. J. and Fedkiw, R. P.: *Level set methods and dynamic implicit surfaces*. Springer-Verlag, New York, 2002.